MEASURING USABILITY FOR APPLICATION SOFTWARE USING THE QUALITY IN USE INTEGRATION MEASUREMENT MODEL

ABDOASSLAM HATAB M KATY

A dissertation submitted in
partial fulfillment of the requirement for the award of the
Degree of Master of Computer Science (Software Engineering)

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

MARCH 2016

To my beloved father and mother

This dissertation is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my beloved mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

"Thank you for your love and support"

ACKNOWLEDGEMENT

First and foremost, I would like to thank almighty God (ALLAH S.W.T) for all the abilities and opportunities. He provided for me through all my life and His blessings enabled me to do this research.

It is the greatest honor to be able to work under the supervision of Pn. Ruhaya Ab Aziz. I am grateful to be one of the luckiest persons who had a chance to work with her. I am thankful and gratified for all of her help, assistance, inspiration, and guidance on all aspects, besides her patience and understanding.

I wish to express my sincere gratitude to everyone who contributed to the successful completion of my study. I would like to express my gratitude to Universiti Tun Hussein Onn Malaysia (UTHM) for all its support throughout my Master's study.

ABSTRACT

User interfaces of application software are designed to make user interaction as efficient and as simple as possible. Market accessibility of any application software is determined by the usability of its user interfaces. A poorly designed user interface will have little value no matter how powerful the program is. Thus, it is significantly important to measure usability during the system development lifecycle in order to avoid user disappointment. Various methods and standards that help measure usability have been developed. However, these methods define usability inconsistently, which makes software engineers hesitant in implementing these methods or standards. The Quality in Use Integrated Measurement (QUIM) model is a consolidated approach for measuring usability through 10 factors, 26 criteria, and 127 metrics. It decomposes usability into factors, criteria, and metrics, and it is a hierarchical model that helps developers with no or little background of usability metrics. Among 127 metrics of QUIM, essential efficiency (EE) is the most specific metric used to measure the usability of user interfaces through an equation. This study involves a comparative analysis between three case studies that use the QUIM model to measure usability in terms of EE for three case studies: (1) Public University Registration System, (2) Restaurant Menu Ordering System, and (3) ATM system. A comparison is made based on the percentage of EE for each element of the use cases in each use case diagram. The results obtained revealed that the user interface design for Restaurant Menu Ordering System scored the highest percentage of EE, thus proving to be the most user-friendly application software among its counterparts.

ABSTRAK

Aspek yang paling penting dalam merekabentuk sesuatu perisian aplikasi adalah menghasilkan antaramuka pengguna yang dapat memastikan interaksi pengguna dan sistem perisian yang ringkas dan efisen. Kebolehpasaran sesuatu perisian aplikasi adalah ditentukan oleh kebolehgunaan antaramuka penggunanya. Antaramuka pengguna yang lemah rekabentuknya menjadi susut nilai kepada sesuatu perisian walau sehebat mana perisian itu dibangunkan. Justeru itu, Pengukuran kebolehgunaan sangat penting untuk dilaksanakan disepanjang kitaran hayat pembangunan sistem untuk memastikan kepuasan hati pengguna. Terdapat pelbagai kaedah dan piawai tentang kebolehgunaan telah di perkenalkan. Namun begitu, kebanyakan kaedah yang digunakan tidak konsisten menyebabkan kebanyakan jurutera perisian menolak untuk mengimplementasikan kaedah tersebut. *Quality in Use Integrated Measurement* (QUIM) kemudiannya diperkenalkan sebagai satu kaedah bersepadu untuk mengukur kebolehgunaan melalui 10 faktor, 26 kriteria, dan 120 metrik. Selain itu, QUIM juga adalah satu model hierarki yang dapat membantu pembangun sistem yang tiada atau kurang mempunyai pengetahuan dan pengalaman mengenai metrik kebolehgunaan. Daripada 127 metrik yang diperkenalkan, *essential efficiency* (EE) adalah satu metrik khusus dan spesifik untuk mengukur kebolehgunaan sesuatu antaramuka pengguna menggunakan kaedah matematik atau satu formula khusus. Kajian ini melibatkan analisis pembandingan yang dibuat melibatkan kepenggunaan QUIM untuk mengukur kebolehgunaan terutamanya EE dalam tiga (3) kajian kes, iaitu: i) Sistem Pendaftaran untuk universiti awam; ii) Sistem Pesanan Makanan untuk sebuah restoran; dan iii) Sistem ATM (mesin juruwang). Perbandingan telah dibuat berdasarkan peratusan EE bagi setiap elemen kes guna yang terdapat dalam setiap rajah kes guna untuk setiap kajian kes. Keputusan kajian menunjukkan yang kajian kes ke ii iaitu sistem pemesanan makan untuk restoran mendapat peratusan tertinggi dari aspek EE, sekaligus membuktikan

yang sistem ini adalah merupakan pens1an aplikasi yang paling mesra penggnna berbanding dengan perisian aplikasi yang lain.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Research Background

User interface is a representation of an application software to the user and communicates with the user through the input fields, pictures, sounds, colors, and text it displays. Even little details in the interface design play a crucial role in creating an impression of overall use. These details elaborate the interaction of the end-user with the application software from the perspective of the user. In today's software, user interfaces are complex and of low-grade quality (Miao *et al.*, 2010). Some software is unnecessarily difficult to comprehend and complex to use. Such software waste the time of users and causes frustration and disappointment in exploring and learning them (Bevan *et al.*, 1994). User interface design will provide effective communication and ease-of-use for both expert and beginner users. The overall degree of use in interface design is referred to as quality in use or usability (Chao, 2009b). ISO 9241-11 (standard related to usability) defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO, 1998).

Different areas are considered in designing a user interface with good usability, such as perception of user, learnability, effectiveness, and user satisfaction. A user perceives a software based on the representation of its functions. Moreover, a user-centered software should help the user learn the system and meet expectations. The value of a software is in its effectiveness. It should provide necessary and effective functions to meet the reasonable needs of the user with minimum time and

effort. In order to achieve user satisfaction, interface designers should consider the users so they can participate from the beginning (Chao, 2009b). For an all-inclusive view, a good quality-in-use model should capture all the features needed for a product to meet predefined usability goals in a particular context of use (Seffah *et al.*, 2001).

Recently, usability measurement has become a major area of study for developing international standards, directives, and theory, as well as empirical research (Seffah *et al.*, 2006). Primarily, the problems in this area are solved by developing and incorporating usability standards, metrics, data, and methods from various resources into a single knowledge base, such as the Quality in Use Integrated Measurement (QUIM), which has a repository containing 10 factors, 26 criteria, and 127 metrics (measureable attributes) for examining the usability of an application software (Padda, 2009). Such frameworks or tools provide the means for a software development team to consider the user perspective on software quality.

Research on software usability measurement has received extensive attention from researchers in human–computer interaction (HCI) communities and from software engineers (Seffah *et al.*, 2001). These researchers have developed various measurement techniques to help establish results in terms of the quality of use of an application software. These techniques, standards, or frameworks are applicable in every stage of a system development lifecycle (SDLC), and they convert customer-oriented characteristics into measureable characteristics. Examples of other usability measurement methods are ISO 9241-11, ISO/TR 16982:2002, ISO/IEC 14598-1, Component IEEE 610, UsabilityNet (a project funded by the European Union), and ESPRIT MUSiC (Molich *et al.*, 2010; Rudisill, 1996). However, each of these standards defines usability differently and emphasizes different sets of usability factors, such as efficiency, effectiveness, learnability, and user satisfaction (Braz *et al.*, 2007).

Moreover, usability without context of use is meaningless. The influential characteristics of the context, such as users, functions, and environment, determine the usability of a software system (Bevan *et al.*, 1994). Consequently, software usability measurement can be broadly categorized into essential and overall usability. To observe essential usability instead of functionality in the overall business process, each software component is contemplated to determine how intact its functionality is

with the ideal use case (Hawkins *et al.*, 2012). That is, the quality of use of an overall application software indicates the general purpose of an application. By contrast, essential functionality in a specific context (use case) (Bevan *et al.*, 1994) is for narrowing down usability (essential usability or essential efficiency).

This research analyzes the essential usability or essential efficiency (EE) of user interfaces of three types of application software related to three case studies using the QUIM model which follows the IEEE 1061 (1998) standard (Software Quality Metrics Methodology). Chapter 2 provides more details on QUIM, and Chapter 4 presents the implementation of this research.

1.2 Problem Statement

There are different types of application software available to help the users to perform specific tasks. Application software can be a word processor, spreadsheet, database software, multimedia software, web based software or any software designed to achieve a specific type of task desired by the user (Stair & Reynolds., 2011). In the application software, the end user directly interacts with the user interface of the application software. Designing user interfaces of application software is a complex undertaking that requires enormous effort to achieve good usability. For an application software to have better usability, the design principles of the user interface must be based on basic understanding of cognitive aspects of Human–computer interaction (HCI). HCI research focuses particularly on the interfaces between people (users) and computers (Pew., 2002). In the domain of HCI, usability studies the elegance and clarity with which the interaction with a computer program (e.g. application software) is designed. Even the nominal details of the interface design contribute to usability and play a vital role in the overall user experience. The major challenges faced by interface designers or software engineers are (a) meeting user expectations, (b) creating a user-friendly design for both beginner and expert users, (c) improving the effectiveness of the system (Constantine, & Lockwood., 1999). To address these challenges, usability methods and standards (i.e., ISO 9241-11, ISO/TR 16982:2002, ISO/IEC 14598-1, Component IEEE 610, UsabilityNet, and ESPRIT MUSiC) must be utilized from the

beginning of the system development lifecycle (SDLC). These standards help measure the usability of an application software from the perspective of the users.

Despite the existence of many individual methods of evaluating usability, software developers are unable to utilize as there is no integrated framework available. Each of these methods perform usability measurement individually, and this creates difficulty for developers to relate the results. To meet this problem, the consolidated model called Quality in Use Integrated Measurement (QUIM) encompasses 10 factors; each relates to specific aspect of usability identified in different standards. These 10 factors are further divided into 26 sub-factors called criteria, which are separated into 127 metrics. QUIM can be used to achieve usability goals effectively by generating usability measurement plans with specific metrics (Seffah *et al.*, 2006). Essential efficiency (EE) is one of the 127 metrics of QUIM model, which represents the specific context through particular use cases, and is measure on how efficiently a software functions with reference to ideal use cases. A failure to measure essential usability results in a failure of the software product (Gray & Salber., 2001).

Therefor this research effectively employs the QUIM model to measure usability in terms of the EE of three types of application software. The analytical results of these case studies are then compared to determine the software application with better usability.

1.3 Project Objectives

This study embarks on the following objectives:

(i) to design the user interfaces of three application software, and

(ii) to compare the EE of the three application software.

1.4 Scope of Project

This research uses the QUIM model to measure the Essential Efficiency (EE) of user interface designs of application software related to three case studies: Public University Registration System (Stumpf *et al.*, 2005). Restaurant Menu and Ordering

System(David, 2008). ATM System(Russell, 2004).This research takes into account that the case studies are three variations of application software.

1.5 Dissertation Outline

This thesis comprises six chapters, including the Introduction and Conclusion chapters. The following are the synopsis of each chapter.

Chapter 1: *Introduction*. Apart from providing an outline of the thesis, this chapter contains an overview of the research background, problem to be solved, objectives to achieve, and scope of the study.

Chapter 2: *Literature Review.* This chapter presents several fundamental concepts related to user interface design and usability measurement. The targeted technique of this research, namely, QUIM, is explained in this chapter. Moreover, the usability measurement techniques applied by previous researchers to solve usability evaluation problems are reviewed in this chapter.

Chapter 3: *Research Methodology.* This chapter discusses the research methodology used to conduct the study systematically. The methodology and metrics used to achieve the objectives of this project are explained in this chapter.

Chapter 4: *Design and Implementation.* This chapter explains the implementation and detailed steps used in this work to employ the QUIM technique in the three case studies.

Chapter 5: *Results and Discussion.* The discussion of the analysis obtained from the experiment and the comparison of the results from the previous chapter are presented in this chapter. The final part of this chapter explains the results achieved.

Chapter 6: *Conclusion.* This chapter concludes the thesis based on the objectives achieved by the project and suggests recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

User interface designs for application software have shifted tremendously in recent years. Currently, the user interfaces of software have become complex and of low-grade quality (Miao *et al.*, 2010). Such interfaces utilize almost all media of communication (i.e., images, videos, text, sound, etc.) to develop an interactive software product. These interfaces define the usability of a software, which, in turn, determines its productivity and acceptance among end-users (Abran *et al.*, 2003). Therefore, measuring the usability or efficiency of the software undergoing the design process is mandatory for the development team. This can be achieved by having specific and predefined quantifiable objectives for usability engineering (Sauro *et al.*, 2005).

Various methods and standards have been developed to measure usability. Nonetheless, none of these approaches covers all aspects of usability because each approach targets different views of usability (Braz *et al.*, 2007). Quality in Use Integrated Measurement (QUIM) is a consolidated method comprising of 10 factors, 26 criteria, and 127 metrics for examining the usability of an application software (Padda, 2009).

This chapter covers the QUIM in extensive detail. The application and advantages of QUIM are also discussed. Other than QUIM, other literature related to usability, such as interface efficiency, software quality, essential usability metrics,

and essential use cases, are also presented in this chapter. As this research focuses on application software, the next section elaborates this type of software in detail.

2.2 Application Software

Application software refers to those computer programs that utilize the capacity of computer to perform specific task. Applications software (also called end-user programs) can be Web/Mobile application, Artificial intelligence software, Product-line software and etc. (Beal., 2010). These applications are programmed to perform specific tasks. There are various types of application program designed to ease the work process of computer users. A user is able to exercise flexibility and perform any task proficiently.

2.2.1 Application Software Domains and Their Characteristics

There are different types of application software utilized to make the task of the user easy (Pressman., 2005; Norton.,1999). The description of some major application software domains as below:

(i) **Word Processing software:** A word processor is a computer software application which performs the tasks of composition, editing, formatting and printing of documents. Today's word processing software include innovations; such as, spell-checking programs and improved formatting options.

(ii) **Spreadsheet Application:** A spreadsheet is an interactive computer application for organizing, analyzing and storing data in tabular form. Spreadsheets are developed as computerized version of accounting worksheets. The program operates on data represented as cells of an array, organized in rows and columns. Each cell of the array may contain either numeric or text data, or the results of formulas that automatically calculate and display a value based on the contents of the other cells.

(iii) **Database management software (DBMS):** RDBMS is a computer program (or more typically a suite of programs) designed to manage a database, a large

set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMS use include accounting, human resources and customer support systems.

(iv) **Graphics, Multimedia Application:** This is a subclass of application software used for graphic design, multimedia development, stylized image development, technical illustration, general image editing, or simply to access graphic files. Art software uses either raster or vector graphic reading and editing methods to create, edit, and view art.

(v) **Engineering / Scientific Software**: A broad array of number crunching programs that range from astronomy to volcanology. From automotive stress analysis to orbital dynamics, and from computer-aided design to molecular biology. from genetic analysis to meteorology.

(vi) **Embedded Software**: This software resides within a product or system and is used to implement and control features and functions for the end user and for the system itself. Embedded software can perform limited and esoteric functions (e.g., key pad control for a microwave oven) or provide significant function and control capability (e.g. digital functions in an automobile such as fuel control, dashboard displays. and braking systems).

(vii) **Product-line Software**: Product-line software is designed to provide a specific capability for use by many different customers. Product-line software can focus on a limited and esoteric marketplace (e.g., inventory control products) or address mass consume.

(viii) **Web/Mobile Applications**: This network-centric software category spans a wide array or applications and encompasses both browser based applications and software that reside on mobile devices.

2.3 Usability and Human Computer Interaction (HCI)

Usability and HCI are becoming core aspects of the system development process to improve and enhance system facilities and to satisfy users' needs and necessities. HCI will assist designers, analysts and users to identify the system needs from text style, fonts, layout, graphics and color, while usability will confirm if the system is efficient, effective, safe, utility, easy to learn, easy to remember, easy to use and to

evaluate, practical visible and provide job satisfaction to the users. Adopting these aspects in the system development process, including the sustainable design will measure and accomplish users' goals and tasks by using a specific technology (Issa & Isaias., 2015). In simple words, usability is defined as the ease of use and learnability of an application software for the end user. Moreover, in further logical terms, usability comprises three quality components: ease of use (EOU) or utility, reliability, and efficiency. These elements define quality in use and are the needs of any user of application software (Speicher, 2015).

Studies have shown the importance of EOU in measuring user satisfaction, which strongly relates to software usability and its acceptance. User interface features associated with EOU also help enhance the learnability and adoptability of the application software among its users (Calisir *et al.*, 2004). Many would argue that EOU is the inverse of complexity or it exists independently of usefulness, and can thus be optimized separately. However, EOU appears different in practice. Research should be conducted to determine the relationship among complexity, usefulness, and EOU (Keil *et al.*, 1995).

Efficiency is another element of usability, which allows the user to perform functions fast and with less effort. This research focuses on measuring usability in terms of efficiency. The following section builds the concept of efficiency in broader detail.

2.4 The Efficiency of User Interface Design

The efficiency of a software system and its interface encompasses a variety of aspects taken together, such as execution time, performance, user satisfaction, and learnability. To achieve the desired goals of accuracy and completion of the task, an efficient user interface should also be able to expend resources easily to the user (Abran *et al.*, 2003; ISO, 1998).

Other fundamental interface design principles in the design of efficient user interfaces are clarity (clear visual elements), flexibility (enabling targeted users with different skill levels to use the interface easily), obviousness (easily learned and understood), availability (make all desired objects available any time), and aesthetically pleasing (provide visual appeal) (Galitz, 2007). Recent advances in

REFERENCES

Abran, A., Khelifi, A., Suryn, W., & Seffah, A. (2003). Usability meanings and interpretations in ISO standards. *Software Quality Journal, 11*(4), 325-338.

Amyot, D., He, X., He, Y., & Cho, D. Y. (2003). Generating Scenarios from Use Case Map Specifications. *QSIC, 3*, 108-115.

Aziz, N. S., Kamaludin, A., & Sulaiman, N. (2013). Assessing web site usability measurement. *International Journal of Research in Engineering and Technology, 2*(9), 386-392.

Basili, V. R., Briand, L. C., & Melo, W. L. (1996). A validation of object-oriented design metrics as quality indicators. *Software Engineering, IEEE Transactions on, 22*(10), 751-761.

Bertoa, M., & Vallecillo, A. (2004). *Usability metrics for software components.* Paper presented at the 8th International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'2004), Oslo, Norway.

Bevan, N. (1999). Quality in use: Meeting user needs for quality. *Journal of Systems and Software, 49*(1), 89-96.

Bevan, N., & Macleod, M. (1994). Usability measurement in context. *Behaviour & information technology, 13*(1-2), 132-145.

Biddle, R., Noble, J., & Tempero, E. (2002). *Essential use cases and responsibility in object-oriented development.* Paper presented at the Australian Computer Science Communications.

Biddle, R., Noble, J., & Tempero, E. (2012). Use Case Cards and Roleplay for Object Oriented Development.

Braz, C., Seffah, A., & M'Raihi, D. (2007). Designing a trade-off between usability and security: A metrics based-model *Human-Computer Interaction–INTERACT 2007* (pp. 114-126): Springer.

Buxton, W. A., Harrison, B. L., & Vicente, K. J. (2000). Graphical user interface with optimal transparency thresholds for maximizing user performance and system efficiency: Google Patents.

Calisir, F., & Calisir, F. (2004). The relation of interface usability characteristics, perceived usefulness, and perceived ease of use to end-user satisfaction with enterprise resource planning (ERP) systems. *Computers in human behavior, 20*(4), 505-515.

Chao, G. (2009a). *Human-computer interaction: process and principles of human-computer interface design.* Paper presented at the Computer and Automation Engineering, 2009. ICCAE'09. International Conference on.

Chao, G. (2009b). *Human-computer interaction: The usability test methods and design principles in the human-computer interface design.* Paper presented at the Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on.

Church, R. L. (1993). *Measuring usability against usability goals during the product development cycle.* Paper presented at the Northcon/93. Conference Record.

Constantine, L. L., & Lockwood, L. A. (1999). *Software for use: a practical guide to the models and methods of usage-centered design*: Pearson Education.

David, C. (2008). Restaurant Menu & Ordering System. from https://www.scribd.com/doc/48168335/restaurant-srs-problem

Finstad, K. (2010). The usability metric for user experience. *Interacting with Computers, 22*(5), 323-327.

Frøkjær, E., Hertzum, M., & Hornbæk, K. (2000). *Measuring usability: are effectiveness, efficiency, and satisfaction really correlated?* Paper presented at the Proceedings of the SIGCHI conference on Human Factors in Computing Systems.

Galitz, W. O. (2007). *The essential guide to user interface design: an introduction to GUI design principles and techniques*: John Wiley & Sons.

Guntupalli, R. C. C. (2008). User interface design: methods and qualities of a good user interface design.

Hagal, M., & Fazzani, F. H. (2012). *A use case map as a visual approach to reduce the degree of inconsistency*. Paper presented at the Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on.

Hassine, J. (2011). *Early availability requirements modeling using use case maps.* Paper presented at the Information Technology: New Generations (ITNG), 2011 Eighth International Conference on.

Hawkins, C., Ross, M., & Staples, G. (2012). *Software Quality Management VI: Quality Improvement Issues*: Springer Science & Business Media.

Hussain, A., & Ferneley, E. (2008). *Usability metric for mobile application: a goal question metric (GQM) approach.* Paper presented at the Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services.

Issa, T., & Isaias, P. (2015). Usability and Human Computer Interaction (HCI). In Sustainable Design (pp. 19-36).

ISO, S. (1998). 9241-11. 1998. *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)–Part II Guidance on Usability.*

Kamalrudin, M., Hosking, J., & Grundy, J. (2011). *Improving requirements quality using essential use case interaction patterns.* Paper presented at the Proceedings of the 33rd International Conference on Software Engineering.

Keil, M., Beranek, P. M., & Konsynski, B. R. (1995). Usefulness and ease of use: field study evidence regarding task considerations. *Decision Support Systems, 13*(1), 75-91.

Mandel, T. (1997). *The elements of user interface design* (Vol. 20): Wiley New York.

McConnell, S. (1993). Code complete: a practical handbook of software construction (Redmond, WA: Microsoft Press.

Miao, Y., & Yang, X. (2010). *An FSM based GUI test automation model.* Paper presented at the Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on.

Molich, R., Chattratichart, J., Hinkle, V., Jensen, J. J., Kirakowski, J., Sauro, J., . . . Traynor, B. (2010). Rent a Car in Just 0, 60, 240 or 1,217 Seconds?-Comparative Usability Measurement, CUE-8. *Journal of Usability Studies, 6*(1), 8-24.

Norton, P. (1999). *Peter Norton's introduction to computers*. Glencoe/McGraw-Hill.pp23-232

Toh, C. K. (2012). *Wireless ATM and Ad-Hoc Networks: Protocols and Architectures*.

Padda, H. (2009). QUIM: A Model for Usability/Quality in use Measurement.

Pressman, R. S. (2005). *Software engineering: a practitioner's approach*: Palgrave Macmillan.

Rauterberg, M. (1996). *How to measure and to quantify usability attributes of man-machine interfaces.* Paper presented at the Robot and Human Communication, 1996., 5th IEEE International Workshop on.

Rudisill, M. (1996). *Human-Computer Interface Design: success stories, emerging methods, and real-world context*: Morgan Kaufmann Pub.

Russell. (2004). Automatic teller machine (ATM). from http://www.mathcs.gordon.edu/courses/cs211/ATMExample/Links.html

Sauro, J., & Kindlund, E. (2005). *How long should a task take? identifying specification limits for task times in usability tests.* Paper presented at the Proceeding of the Human Computer Interaction International Conference (HCII 2005), Las Vegas, USA.

Schneidewind, N. F. (1992). Methodology for validating software metrics. *Software Engineering, IEEE Transactions on, 18*(5), 410-422.

Seffah, A., Donyaee, M., Kline, R. B., & Padda, H. K. (2006). Usability measurement and metrics: A consolidated model. *Software Quality Journal, 14*(2), 159-178.

Seffah, A., Kececi, N., & Donyae, M. (2001). *QUIM: a framework for quantifying usability metrics in software quality models.* Paper presented at the Quality Software, 2001. Proceedings. Second Asia-Pacific Conference on.

Speicher, M. (2015). What is Usability?

Stumpf, R. V., & Teague, L. C. (2005). Object-oriented analysis and design: Tsinghua University Press, Beijing, China. pp.

Venkatesan, V. P., & Krishnamoorthy, M. (2009). A metrics suite for measuring software components. *Journal of Convergence Information Technology, 4*(2), 138-153.

Winter, J., Rönkkö, K., Ahlberg, M., Hinely, M., & Hellman, M. (2007). *Developing Quality through Measuring Usability--The UTUM Test Package.* Paper presented at the Software Quality, 2007. WoSQ'07: ICSE Workshops 2007. Fifth International Workshop on.

Yamada, S., Hong, J.-K., & Sugita, S. (1995). Development and evaluation of hypermedia for museum education: validation of metrics. *ACM Transactions on Computer-Human Interaction (TOCHI), 2*(4), 284-307.

Yang, Z., Yang, S., Yu, Z., Yin, B., & Bai, C. (2015). Graphical User Interface Reliability Prediction Based on Architecture and Event Handler Interaction. In W. E. Wong (Ed.), *Proceedings of the 4th International Conference on Computer Engineering and Networks*: Springer International Publishing,pp. 1003-1010.